

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Experiment 3

Student Name: Umang Kumar
Branch: BE- CSE(AIML)
Semester: 6th
Subject Name: Full Stack

UID: 23BAI70024
Section: 23AIT KRG 1(G2)
Date of Performance: 8/02/26
Subject Code: 23CSH-382

- 1. Aim:** To optimize the performance of the EcoTrack React application using memoization techniques and code splitting, and to enhance the user interface using enterprise-grade Material UI components.

2. Objective:

After completing this experiment, the student will be able to:

- Understand the causes of unnecessary re-renders in React applications
- Optimize React components using `React.memo` to prevent avoidable re-renders
- Apply `useMemo` to efficiently compute derived data and avoid redundant calculations
- Use `useCallback` to memoize event handler functions and improve component performance
- Implement lazy loading of components and routes using `React.lazy` and `Suspense`
- Reduce initial bundle size and improve application load performance through code splitting
- Enhance the visual appearance and usability of the EcoTrack application using Material UI components
- Design a clean, consistent, and responsive user interface using Material UI layouts and typography

3. Implementation/Code:

Header.js:

```
import {Link} from "react-router-dom";  
  
const Header = ({ title }) => {  
  return (  
    <div>
```

```

<header style={ {
    padding:"1rem",
    backgroundColor:"#27ae60",
    color:"white",
    textAlign:"center",
}}>
    <h2>{title}</h2>
    <nav>
        <Link to="/" style={{marginRight:"1rem",color:"#79e44f"}}>
            Dashboard
        </Link>
        <Link to="/logs"
style={{marginRight:"1rem",color:"#79e44f"}}>
            Logs
        </Link>
        <Link to="/login"
style={{marginRight:"1rem",color:"#79e44f"}}>
            Login
        </Link>

        <Link to="/logout"
style={{marginRight:"1rem",color:"#79e44f"}}>
            Logout
        </Link>
    </nav>
</header>
);
};

export default Header;

```

AuthContext.js:

```

import { createContext,useContext,useState } from "react";

const AuthContext = createContext(null);

export const AuthProvider =({children})=>{

```

```
const [ isAuthenticated, setIsAuthenticated ] = useState(false);

return(
  <AuthContext.Provider value={ { isAuthenticated,
    setIsAuthenticated } }>
    {children}
  </AuthContext.Provider>
)
}
```

```
export const useAuth = () => useContext(AuthContext);
```

logs.js:

```
export const logs = [
  { id: 1, activity: "Car Travel", carbon: 4 },
  { id: 2, activity: "Electricity Usage", carbon: 6 },
  { id: 3, activity: "Cycling", carbon: 0 },
  { id: 4, activity: "Bus Travel", carbon: 3 },
  { id: 5, activity: "Solar Energy Usage", carbon: 1 },
  { id: 6, activity: "Flight Travel", carbon: 8 },
];
```

```
export default logs;
```

Dashboard.js:

```
import logs from './data/logs'
import Header from './components/header'
const totalCarbon = logs.reduce ((sum, log) => sum+log.carbon,0)

const Dashboard = () => {
  return (
    <div >
      <h2>{Header}</h2>
      <p style={{padding
        :"1rem",backgroundColor:"#3175a2",color:"#d6eaf8",textAlign:"cente
        r" }}>Total Carbon Footprint: {totalCarbon}</p>
      <ul style={{padding
```

```

        :"1rem",backgroundColor:"white",color:"#111",textAlign:"center"}}>
    {logs.map((log)=>(
        <li key ={log.id}>
            {log.activity}:{log.carbon} kgs
        </li>
    )))
</ul>
</div>

)
}

export default Dashboard;

```

DashboardAnalytics.js:

```

const DashboardAnalytics =()=>{
    return (
        <p>This is Dashboard Analytics</p>
    )
};

export default DashboardAnalytics;

```

DashboardSummary.js:

```

const DashboardSummary =()=>{
    return (<p>This is Dashboard Summary</p>)
}

export default DashboardSummary;

```

DashboardSettings.js:

```

const DashboardSettings = () => {
    return (<p>These are the settings</p>)
}

export default DashboardSettings;

```

DashboardLayout.js:

```

import { Link,Outlet } from "react-router-dom";

```

```

const DashboardLayout = ()=>{
  return (
    <div style={{padding:"1rem"}}>
      <h3>Dashboard</h3>

      <nav>
        <Link to="/summary">Summary</Link> | {" "}
        <Link to="/analytics">Analytics</Link>
      </nav>
      <hr/>
      <Outlet/>

    </div>
  );
}

export default DashboardLayout;

```

Login.js:

```

import { useAuth } from "../context/AuthContext";
import { useNavigate } from "react-router-dom";

const Login = ()=>{
  const { setIsAuthenticated } = useAuth();
  const navigate = useNavigate();

  const handleLogin = () => {
    setIsAuthenticated(true);
    navigate("/");
  };

  return (
    <div style={{padding:"1rem"}}>
      <h3>Login Page</h3>
      <button onClick={handleLogin}>Login</button>
    </div>
  );
}

```

```
export default Login;
```

Logout.js:

```
import { useAuth } from "../context/AuthContext";
import { useNavigate } from "react-router-dom";

const Logout = () => {
  const { setIsAuthenticated } = useAuth();
  const navigate = useNavigate();

  const handleLogout = () => {
    setIsAuthenticated(false);
    navigate("/");
  };

  return (
    <div style={{ padding: "1rem" }}>
      <h3>Logout Page</h3>
      <button onClick={handleLogout}>Logout</button>
    </div>
  )
}

export default Logout;
```

ViewLogs.js:

```
import logs from "../data/logs";

const Viewlogs = () => {
  return (
    <div>
      <h2>Experiment Logs</h2>
      <ul>
        {logs.map((log, index) => (
          <li key={log.id}>
            {log.activity}: {log.carbon} kg CO2
          </li>
        ))}
      </ul>
    </div>
  )
}
```

```
}

export default Viewlogs;
```

ProtectedRoute.js:

```
import {Navigate} from "react-router-dom";
import { useAuth } from "../context/AuthContext";

const ProtectedRoute= ({children})=>{
  const {isAuthenticated} = useAuth();

  if(!isAuthenticated){
    return <Navigate to="/login" replace/>
  }

  return children;
}

export default ProtectedRoute;
```

LogSlice.js:

```
import { createSlice,createAsyncThunk } from "@reduxjs/toolkit";

export const fetchLogs = createAsyncThunk(
  "logs/fetchLogs",
  async()=> {
    await new Promise ((resolve) => setTimeout(resolve,1000));
    return [
      { id: 1, activity: "Car Travel", carbon: 4 },
      { id: 2, activity: "Electricity Usage", carbon: 6 },
      { id: 3, activity: "Cycling", carbon: 0 },
      { id: 4, activity: "Bus Travel", carbon: 3 },
      { id: 5, activity: "Solar Energy Usage", carbon: 1 },
      { id: 6, activity: "Flight Travel", carbon: 8 },
    ];
  }
)
```

```
const logsSlice = createSlice({
  name: "logs",
  initialState: {
    data:[],
  }
```

```

    status:"idle",
    error:null,
},
reducers: {},
extraReducers :(builder)=>{
  builder.addCase(fetchLogs.pending,(state)=>{
    state.status="loading";
  })
  .addCase(fetchLogs.fulfilled,(state,action)=>{
    state.status ="success";
    state.data = action.payload;
  })
  .addCase(fetchLogs.rejected,(state,action)=>{
    state.status = "failed";
    state.error = action.error.message;
  })
}
})
}

export default logsSlice.reducer;

```

Store.js:

```

import {configureStore} from "@reduxjs/toolkit"
import logsReducer from "./logsSlice";

const store = configureStore({
  reducer :{
    logs: logsReducer,
  },
});

export default store;

```

App.js:

```

import Header from "./components/header";
import { Suspense, lazy } from "react";

```

```
import { BrowserRouter, Routes, Route } from "react-router-dom";

const Login = lazy(()=>import("./pages/login"))
const DashboardLayout = lazy(()=> import("./pages/DashboardLayout"))
const Viewlogs = lazy(()=>import("./pages/Viewlogs"))
const DashboardAnalytics =
    lazy(()=>import("./pages/DashboardAnalytics"))
const DashboardSummary =
    lazy(()=>import("./pages/DashboardSummary"))
const ProtectedRoute = lazy(()=>import("./routes/ProtectedRoutes"))
const Logout = lazy(()=>import("./pages/logout"))
const DashboardSettings =
    lazy(()=>import("./pages/DashboardSettings"))

function App (){
    return (
        <BrowserRouter>
            <Header title="Eco Track : Experiment 2 "/>
            <Suspense fallback={<h2>Loading...</h2>}>
                <Routes>
                    <Route path= "login" element={<Login/>}/>
                    <Route path= "/" element=
                    {
                        <ProtectedRoute>
                            <DashboardLayout />
                        </ProtectedRoute>
                    }>
                    <Route index />
                    <Route path="settings" element={<DashboardSettings />} />
                    <Route path="Summary" element={<DashboardSummary/>} />
                    <Route path="Analytics" element={<DashboardAnalytics/>} />
                    <Route path="logs" element={<Viewlogs/>} />
                </Routes>
            </Suspense>
        </BrowserRouter>
    )
}
```

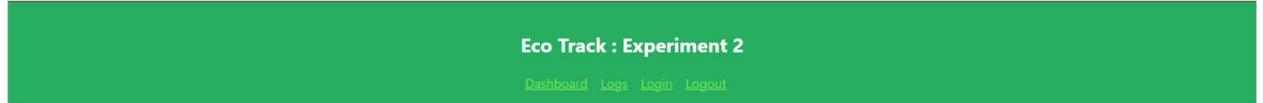
```
</Suspense>

</BrowserRouter>
)
}

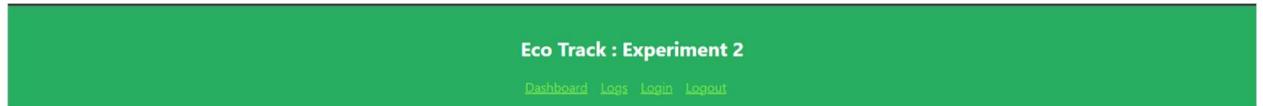
export default App;
```

4. Output:

Login page:



Dashboard Page:



Dashboard Summary Page:

Eco Track : Experiment 2

[Dashboard](#) [Logs](#) [Login](#) [Logout](#)

Dashboard

[Summary](#) | [Analytics](#)

This is Dashboard Summary

localhost:3000/summary

Logs Page:

Eco Track : Experiment 2

[Dashboard](#) [Logs](#) [Login](#) [Logout](#)

Dashboard

[Summary](#) | [Analytics](#)

Experiment Logs

- Car Travel: 4 kg CO2
- Electricity Usage: 6 kg CO2
- Cycling: 0 kg CO2
- Bus Travel: 3 kg CO2
- Solar Energy Usage: 1 kg CO2
- Flight Travel: 8 kg CO2

Logout Page:

Eco Track : Experiment 2

[Dashboard](#) [Logs](#) [Login](#) [Logout](#)

Logout Page

[Logout](#)

5. Learning Outcome

- 1. Learnt about react files.**
- 2. Learnt Design and apply redux and its uses.**
- 3. Learnt Implement nested routing to build dashboard-style layouts.**